

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently Amended) A computer program product encoding a computer program for executing on a computer system a computer process for generating a customized library for execution of an application by a client, the client including one or more client-loaded types, the computer process comprising:

identifying one or more application-referenced types developed by third parties on which the application depends for execution but which are not part of the application;

identifying one or more client-needed types required by the client to execute the application, based on the application-referenced types and the client-loaded types;

generating the customized library, including the one or more client-needed types; and separately sending the customized library and the application to the client.

2. (Original) The computer program product of claim 1 wherein the operation of generating a customized library comprises:

inserting only client-needed types into the customized library.

3. (Original) The computer program product of claim 1 wherein the operation of generating a customized library comprises:

inserting all client-needed types and all client-loaded types into the customized library.

4. (Original) The computer program product of claim 1 wherein the operation of identifying one or more application-referenced types comprises:

identifying a version identifier of at least one of an application-referenced type and a client-needed type.

5. Cancelled

6. (Original) The computer program product of claim 1 wherein operation of generating a customized library comprises:

excluding from the customized library one or more non-identified types, each non-identified type not being referenced by the application.

7. (Original) The computer program product of claim 1 wherein the operation of identifying one or more application-referenced types comprises:

examining the application to identify one or more type references, each type reference being associated with an application-referenced type; and

generating a dependency list identifying the one or more application-referenced types, based on the one or more type references.

8. (Original) The computer program product of claim 7 wherein the operation of generating a dependency list comprises:

recursively examining an application-referenced type to identify one or more additional type references referenced by the application-referenced type; and

adding the one or more additional type references to the dependency list.

9. (Original) The computer program product of claim 1 wherein the operation of identifying one or more client-needed types comprises:

identifying the one or more client-loaded types; and

generating a client composite list to identify the one or more client-needed types, the one or more client-needed types including the one or more application-referenced types not loaded on the client.

10. (Original) The computer program product of claim 9 wherein the operation of generating a client composite list comprises:

• determining whether a new version of an application-referenced type is needed by the client, based on a versioning policy.

11. (Original) The computer program product of claim 9 wherein the operation of identifying one or more client-loaded types comprises:

accessing a client catalog specifying the one or more client-loaded types loaded on the client; and

examining the client catalog to identify the one or more client-loaded types.

12. (Original) The computer program product of claim 1 wherein the operation of identifying one or more client-needed types comprises:

identifying the one or more client-loaded types by receiving a list of the client-loaded types from the client; and

evaluating the client-loaded types against the application-referenced types to identify the client-needed types.

13. (Original) The computer program product of claim 1 wherein the operation of identifying one or more client-needed types comprises:

receiving a client composite list specifying the one or more application-referenced types not loaded on the client.

14. (Original) The computer program product of claim 1 wherein the operation of generating a customized library comprises:

creating a new library containing no types; and

adding each of the one or more client-needed types to the new library to provide the customized library.

15. (Original) The computer program product of claim 14 wherein the adding operation comprises:

adding one or more global data fields of each client-needed type to the new library.

16. (Original) The computer program product of claim 14 wherein the adding operation comprises:

adding a method signature corresponding to a method of each client-needed type to the

new library; and

adding a code body associated with the method of each client-needed type to the new library.

17. (Original) The computer program product of claim 16 wherein the operation of adding a method signature comprises:

adding to the new library metadata describing the method.

18. (Original) The computer program product of claim 1 wherein an application-referenced type includes a class.

19. (Original) The computer program product of claim 1 wherein an application-referenced type includes at least one of a method, a class, a data field, and a resource.

20. (Original) The computer program product of claim 1 wherein a client-loaded type includes at least one of a method, a class, a data field, and a resource.

21. (Original) The computer program product of claim 1 wherein the computer process further comprises:

receiving a device profile specifying a characteristic of the client; and

excluding an attribute from the customized library, if the attribute is incompatible with the characteristic of the client.

22. (Original) The computer program product of claim 1 wherein the computer process further comprises:

receiving a device profile specifying a characteristic of the client;

identifying more than one client-needed type candidate, based on the client-needed types identified in the operation of identifying one or more client-needed types; and

selecting one of the client-needed type candidates based on the characteristic identified in the device profile to provide one of the client needed types.

23. Cancelled

24. (Currently Amended) A customized library management system for managing a customized class library, the system comprising:

an application server module receiving an application request and transmitting a requested application to a client; and

a customized library generator creating a customized library and separately sending the customized library to the client, the customized library including one or more client-needed types, which are application-referenced types developed by third parties not loaded on the client but required by the application to execute.

25. Cancelled

26. (Currently Amended) A customized library management system for separately receiving a customized library associated with an application, the system comprising:

a catalog identifying one or more client-loaded types that are loaded in a client;

a filter module comparing the catalog to a list of one more application-referenced types, which are developed by third parties but required by the application to execute, and generating a client composite list to identify one or more client-needed types; and

a client receipt module for separately receiving the customized library and the application ~~including the one or more client-needed types.~~

27. (Original) The customized library management system of claim 26 further comprising:

a client request module requesting the application from a server.

28. Cancelled

29. (Original) The customized library management system of claim 26 wherein the catalog comprises:

an application identifier associated with the application loaded on the client.

30. (Original) The customized library management system of claim 26 wherein the catalog further comprises:

dependency information identifying a client-loaded type referenced by the application.

31. (Original) The customized library management system of claim 26 wherein the catalog further comprises:

an install point indicator associated with the application.

32. (Original) The customized library management system of claim 26 wherein the catalog comprises:

a type identifier associated with a client-loaded type; and

dependency information associated with the type identifier and identifying one or more client-loaded applications referencing the client-loaded type.

33. (Original) The customized library management system of claim 26 wherein the catalog comprises:

dependency information recording a reference count of one or more applications referencing the client-loaded type.

34. (Currently Amended) A computer program product encoding a computer program for executing on a computer system a computer process for creating a customized library, the computer process comprising:

accessing a class library store;

identifying one or more needed classes referenced by the application but not included with the application;

creating an empty library;

extracting the one or more needed classes from the class library store; and

adding to the empty library metadata describing the method, the metadata being extracted from the class library store in the extracting operation; and

_____ adding the one or more needed classes to the empty library to generate the customized library.

35. (Original) The computer program product of claim 34 wherein the identifying operation comprises:

receiving a composite list specifying the needed classes.

36. (Original) The computer program product of claim 34 wherein the adding operation comprises:

adding one or more global data fields of each needed class to the empty library, the global data fields being extracted from the class library store in the extracting operation.

37. (Original) The computer program product of claim 34 wherein the adding operation comprises:

adding a method signature corresponding to a method of each needed class to the empty library, the method signature being extracted from the class library store in the extracting operation; and

adding a code body associated with the method of each needed class to the empty library, the code body being extracted from the class library store in the extracting operation.

38. Cancelled

39. (Original) The computer program product of claim 34 wherein the computer process further comprises:

identifying one or more needed methods associated with each needed class.

40. (Previously Presented) The computer program product of claim 34 wherein the computer process further comprises:

identifying one or more needed data fields associated with each needed class.

41. Cancelled

42. (New) A computer resource management system for managing computer resources, the system comprising:

a library storage region that stores at least one library;

an application storage region that stores at least one application;

a catalog that monitors the least recently used application in the application storage region; and

a deflation module that deflates the least recently used application or one or more least recently used libraries if computer resources are limited.

43. (New) A computer resource management system defined in claim 42, wherein the catalog stores an indication of which applications and libraries are deflated.

44. (New) A computer resource management system defined in claim 42, wherein at least some code is deleted when an application or library is deflated.

45. (New) A computer resource management system defined in claim 42, wherein an indicator of the install point for where to request the application is maintained for the application after deflation.

46. (New) A computer resource management system defined in claim 45, wherein the install point is a Uniform Resource Locator of a resource that provides the application or application-referenced types.

47. (New) A computer resource management system defined in claim 42, further comprising:
a regeneration module that regenerates a deflated application when the application is invoked.

48. (New) A computer resource management system defined in claim 47, wherein the regeneration module sends an application request to an install point in a catalog.

49. (New) A computer resource management system defined in claim 48, wherein the regeneration module receives the application and one or more client-needed types to reinstall the deflated application.